

# Radiateur connecté

Lors de cet atelier, Alexis propose la réalisation d'un système de contrôle connecté d'un radiateur électrique. Au menu : un peu d'arduino ou de raspberry pi, de la mesure de température, une touche de communication ethernet ou wifi et un des relais ! Le tout permet par exemple d'économiser un peu d'énergie en programmant une (vieuse) installation.

## Contrôle du radiateur

On positionne le radiateur en mode manuel et on met au maximum. Puis on branche le radiateur à travers une prise électrique commandée par radio. De cette façon, on court-circuite le thermostat interne et la prise contrôle directement la résistance du radiateur.



Pour tester le bon fonctionnement de cette partie, il suffit d'utiliser la télécommande fournie avec la prise: à l'arrêt le radiateur est éteint, en marche le radiateur doit chauffer au maximum sans arrêt.

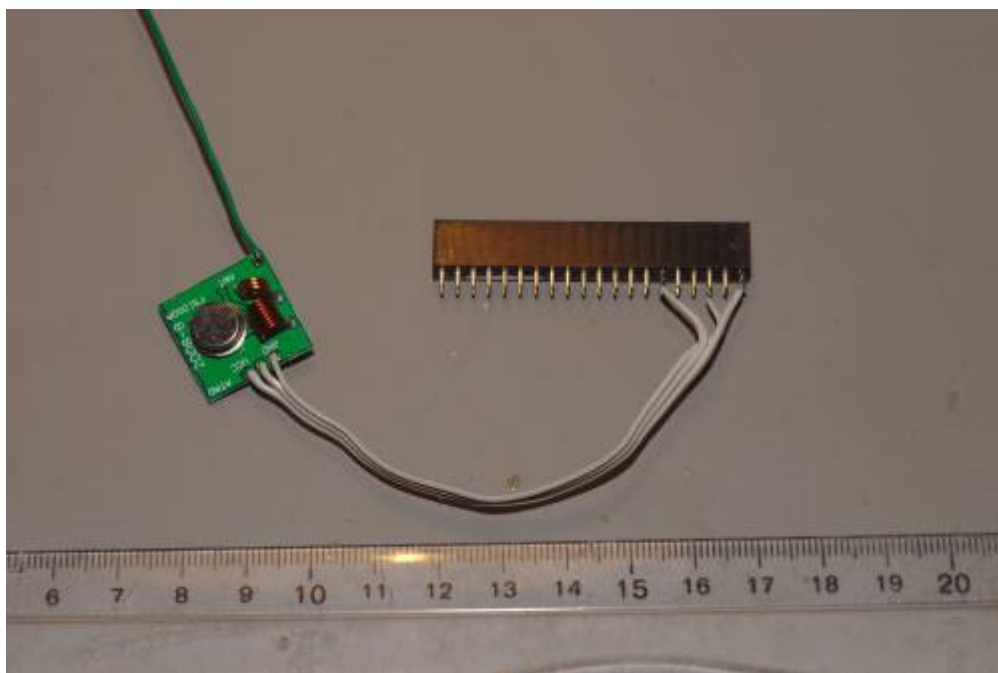


## Contrôle avec le Raspberry Pi

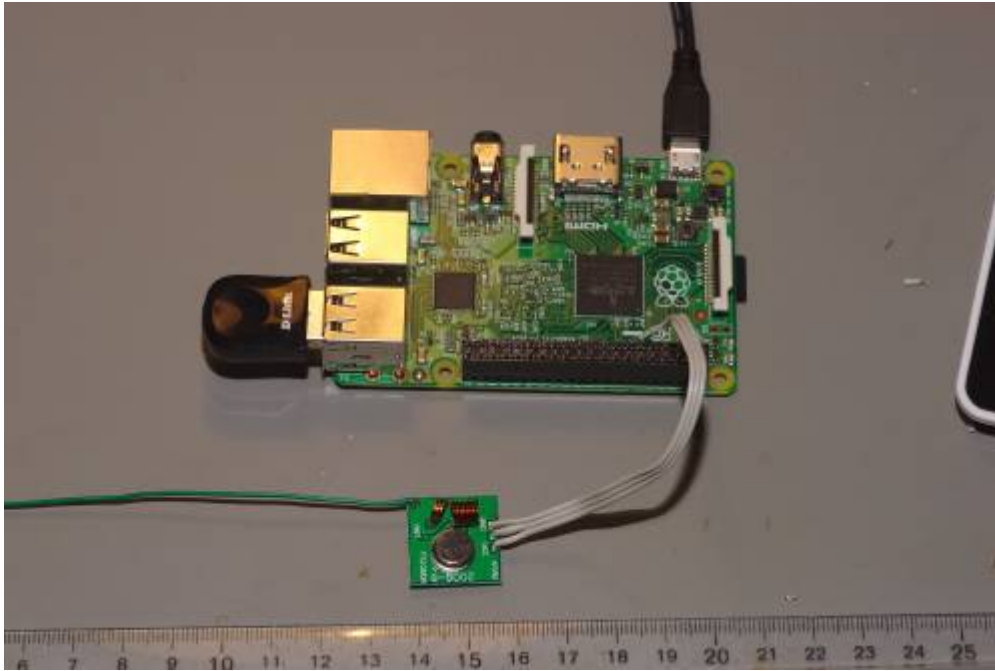
Maintenant que le radiateur peut-être commandé par radio, on va le piloter depuis un programme qui tourne sur le raspberry pi.

On pourrait juste démonter la télé-commande et brancher des pins du raspberry pi à la place des boutons marche/arrêt, mais ce ne serait pas très élégant.

Il existe des modules émetteurs 433 MHz à très bas coût. Ils ont seulement trois pins: deux pour l'alimentation et un pour activer ou désactiver l'oscillateur. Seule la modulation OOK est donc supportée, mais c'est suffisant pour commander des prises DI-O.

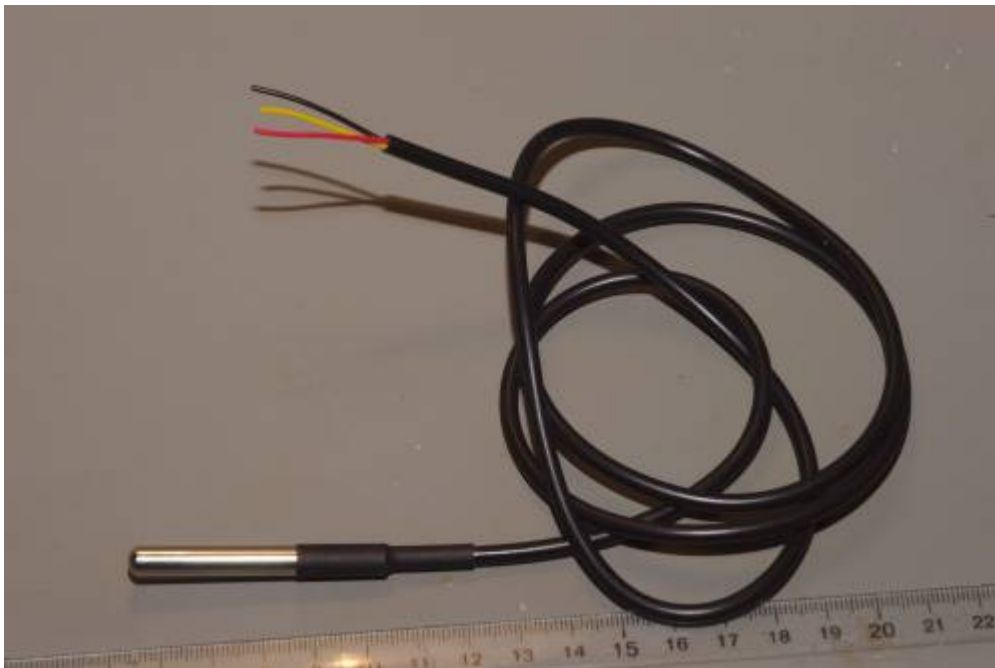


Le branchement de l'émetteur et le logiciel pour le piloter sont détaillés à l'adresse:  
<http://blog.idleman.fr/raspberry-pi-12-allumer-des-prises-distance/>



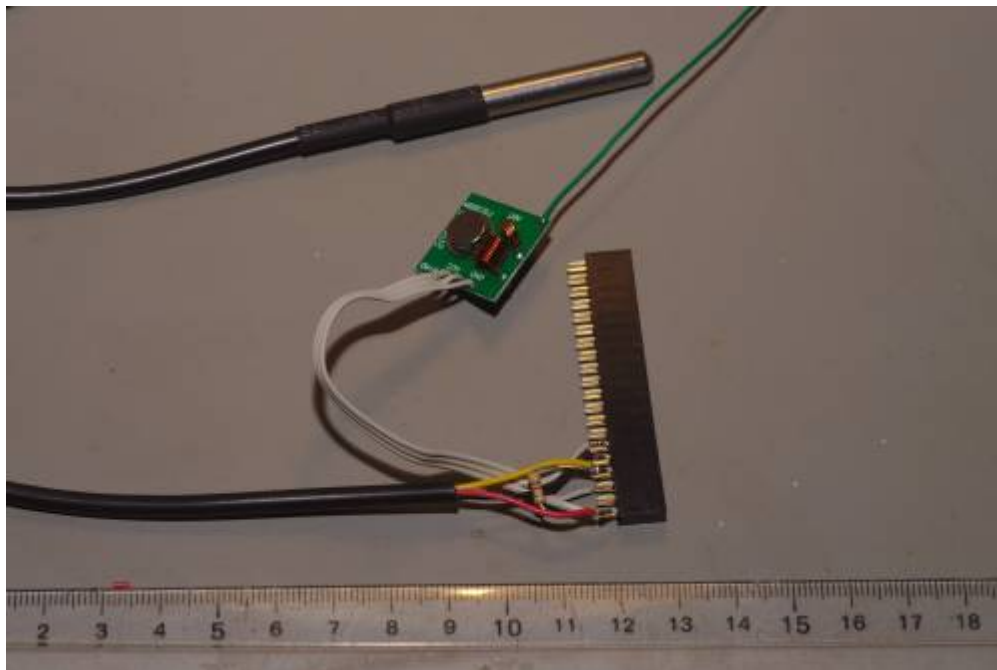
## Mesure de la température avec le Raspberry Pi

On utilise une sonde de température DS18B20 qui renvoie directement la température sous forme numérique.



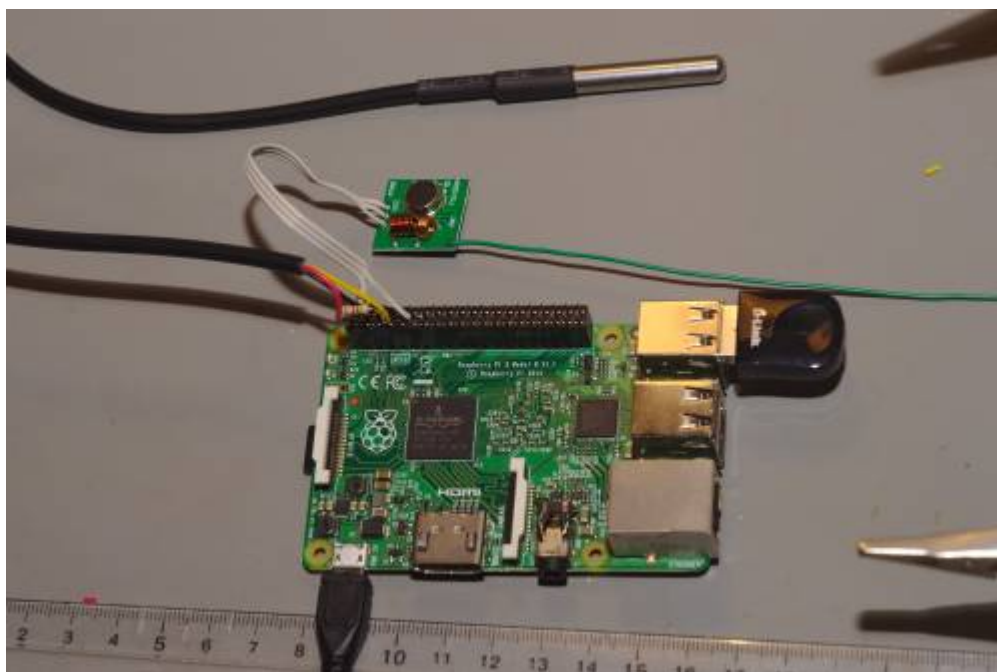
Le branchement est détaillé à l'adresse suivante:

<http://www.framboise314.fr/mesure-de-temperature-1-wire-ds18b20-avec-le-raspberry-pi/>



Le driver pour ces sondes est directement présent dans le noyau Linux et les valeurs sont lisibles depuis un script shell en interrogeant simplement un fichier du sysfs.

## Tout ça mis ensemble pour faire un asservissement de la température



```
#!/bin/bash

get_temp() {
    cat /sys/bus/w1/devices/28-80000028279c/w1_slave | grep -o 't=.*$' | sed
's/t=//'
}

heater_control() {
    sudo /home/pi/dio/hcc/radioEmission 0 12325261 1 $1 >/dev/null
}
```

```
}

target_t=30000

while true; do
    t=$(get_temp)
    echo "t=$t target_t=$target_t"
    sleep 2
    if [ "$t" -lt 15000 -o "$t" -gt 40000 ] ; then
        echo "bad value"
        continue
    fi
    if [ "$t" -lt "$target_t" ] ; then
        echo "heater on"
        heater_control on
    else
        echo "heater off"
        heater_control off
    fi
done
```

## Interface web

On modifie légèrement le script de contrôle pour écrire la température courante dans un fichier et lire la consigne depuis un fichier.

```
#!/bin/bash

get_temp() {
    cat /sys/bus/w1/devices/28-80000028279c/w1_slave | grep -o 't=.*$' | sed
's/t=//'
}

heater_control() {
    sudo /home/pi/dio/hcc/radioEmission 0 12325261 1 $1 >/dev/null
}

while true; do
    target_t=$(cat target_t)
    t=$(get_temp)
    echo "$t" > current_t
    echo "t=$t target_t=$target_t"
    sleep 2
    if [ "$t" -lt 15000 -o "$t" -gt 40000 ] ; then
        echo "bad value"
        continue
    fi
    if [ "$t" -lt "$target_t" ] ; then
        echo "heater on"
```

```
        heater_control on
    else
        echo "heater off"
        heater_control off
    fi
done
```

En parallèle on lance notre interface web.

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def main():
    t = float(open('current_t').read())/1000
    target_t = float(open('target_t').read())/1000
    return 't=%.2f target_t=%.2f' % (t, target_t)

@app.route('/s/<int:new_t>')
def s(new_t):
    f = open('target_t', 'w')
    f.write("%d" % (new_t*1000))
    f.close()
    return 'target_t=%d' % new_t

if __name__ == "__main__":
    app.run(host='0.0.0.0')
```

La page "/" permet d'afficher la température et la page "/s/<nombre>" permet de définir à la consigne à "nombre".

From:

<https://labovilleurbanne.fr/dokuwiki/> - **DokuWiki du LOV**

Permanent link:

[https://labovilleurbanne.fr/dokuwiki/ateliers:radiateur\\_connecte](https://labovilleurbanne.fr/dokuwiki/ateliers:radiateur_connecte)

Last update: **2023/06/11 14:42**

